UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/827,739	04/06/2001	John H. Sandham		6702

36183 7590 08/02/2005

PAUL, HASTINGS, JANOFSKY & WALKER LLP
P.O. BOX 919092
SAN DIEGO, CA 92191-9092

EXAMINER	
PROCTOR, JASON SCOTT	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 08/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/827,739	SANDHAM, JOHN H.	
	Examiner	Art Unit	
	Jason Proctor	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 May 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

5.00

DETAILED ACTION

Claims 1-12 were presented for examination. Claims 1-12 were rejected in Office Action mailed November 19, 2004. Claims 1 and 9 have been amended and new claim 13 added in Applicants' response mailed on May 23, 2005. Claims 1-13 have been rejected.

Priority

1. The Examiner thanks Applicants for supplying a certified copy of the Great Britain application 9822074.2 filed on October 10, 1998 as required by 35 U.S.C. § 119. For the purposes of examination, the priority date of the instant application is regarded as October 10, 1998.

Response to Rejections under 35 U.S.C. § 101

2. Regarding the previous rejections of claims 1-4 under 35 U.S.C. § 101, Applicants' arguments are primarily directed toward the patentability of a method performed by a person. These arguments have been fully considered but have been found unpersuasive. However, upon reconsideration, the phrase "a method for emulating [...] on a second processor," recited by each of these claims, restricts their interpretation to the technological arts. The broadest reasonable interpretation for those claims does not include a human being performing the method. The Examiner acknowledges that these claimed inventions have an asserted utility in the technological arts. The rejection of claims 1-4 under 35 U.S.C. § 101 as being directed to nonstatutory subject matter is therefore withdrawn.

3. Regarding the rejection of claims 5-12, Applicants have argued primarily that:

> In the present situation, the asserted fact that applicants operations may be performed by a person, even though capable of being performed by a processor, does not render the subject matter nonstatutory.

The Examiner respectfully traverses this argument as follows.

The previous Office Action established the rationale for rejecting these claims under 35 U.S.C. § 101:

> [t]he claimed limitations similarly relate to performing an arithmetic memory address translation and recite no tangible embodiment such that a human being would not infringe the claimed limitations. (previous Office Action, paragraph 6)

The Examiner's rationale is supported by MPEP 2106 (IV)(B)(1), which states:

> If the "acts" of a claimed process manipulate only numbers, abstract concepts or ideas, or signals representing any of the foregoing, the acts are not being applied to appropriate subject matter. Schrader, 22 F.3d at 294-95, 30 USPQ2d at 1458-59. Thus, a process consisting solely of mathematical operations, i.e., converting one set of numbers into another set of numbers, does not manipulate appropriate subject matter and thus cannot constitute a statutory process.

Claims 5-12 are methods that convert one set of numbers (memory access addresses in a first endian format) into another set of numbers (memory access addresses in a second endian format). These claims are therefore nonstatutory. Further, as exemplified in claim 12, this is a method of simple arithmetic *"A-B-L+S"*, and anticipated by steps of human cognition given the broadest reasonable interpretation. Further, claim 5 is directed to "compiling or translating a computer program code instruction" while claims 9 and 10 comprise "means for transforming a memory access address". This simplified version of the method further demonstrates that human mental processes would infringe upon the claimed method.

Further, MPEP 2111 instructs that claims are given their broadest reasonable interpretation during examination:

> During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." *In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). Applicant always has the opportunity to amend the claims during prosecution, and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. *In re Prater*, 415 F.2d 1393, 1404-05, 162 USPQ 541, 550-51 (CCPA 1969)

Claims 5-8 are directed toward a disembodied process that does not require the use of any tangible hardware. Claims 9 and 10 are directed to a system defined in means-plus-function language and therefore must be broadly interpreted as similar in scope to the process of claim 5. Although claims 11 and 12 refer to a processor of a first type and a processor of a second type, neither processor is integral to or contributes functionality to the systems. These claimed systems are limited by the method they perform. Therefore, granting the broadest reasonable interpretation to claims 5-11, these claims recite mathematical operations that convert one set of numbers into another set of numbers and are therefore nonstatutory.

4. Applicant further offers arguments that the invention produces

> "the useful and tangible result of transforming the memory access addresses such that bytes stored in memory addressed by a processor of a first endian format are observed as if the memory was addressed by a processor of a second endian format. These address transformations produce the useful, concrete, and tangible result of ensuring that bytes aggregate in the memory in a pattern which is a mirror image of the pattern which would have resulted in the processor had been the other endian format and no address transformation had been performed."

The Examiner respectfully traverses this argument as follows.

Applicants' arguments fail to indicate the *tangible* result of these methods. A "transformed memory access address" is not a tangible result. Again, the Examiner refers to MPEP 2106 (IV)(B)(1):

> If the "acts" of a claimed process manipulate only numbers, abstract concepts or ideas, or signals representing any of the foregoing, the acts are not being applied to appropriate subject matter. Schrader, 22 F.3d at 294-95, 30 USPQ2d at 1458-59. Thus, a process consisting solely of mathematical operations, i.e., converting one set of numbers into another set of numbers, does not manipulate appropriate subject matter and thus cannot constitute a statutory process.

The claimed methods manipulate numbers and do not produce a tangible result. These claims are directed to nonstatutory subject matter.

Response to Provisional Double Patenting Rejections

5. The previous double patenting rejections are hereby withdrawn.

Response to Rejections under 35 U.S.C. § 112, first paragraph

6. The Examiner thanks Applicants for clarifying remarks and citation of the specification where support for the claim limitations is found. The previous rejections of claims 1-4 and 9-12 under 35 U.S.C. § 112, first paragraph, have been withdrawn.

Response to Rejections under 35 U.S.C. § 112, second paragraph

7. Regarding the rejection of claims 1-4 under 35 U.S.C. § 112, second paragraph, the Examiner thanks Applicant for clarifying remarks. Applicants argue primarily that:

> [...] the meaning of the term "emulating" appearing in the claims and specification of the present application is commensurate with the meaning understood to those skilled in the art. In fact, the dictionary definition provided by the Examiner from the Microsoft Computer Dictionary fully coincides with the definition provided by the Applicant in the present specification.

Applicants' argument attempts to correlate the definition of *emulate* as a verb to the definition of "emulation systems" as found in the specification (page 1, lines 16-18).

The Examiner respectfully traverses this argument as follows.

The Examiner respectfully draws Applicants' attention to the definition of *emulate* as provided by Applicants:

> **Emulate** vb. For a hardware or software system to behave in the same manner as another hardware [or] software system.

The specification (page 1, lines 22-24, cited by Applicants) shows that the invention *enables* emulation to be performed and thus distinguishes the invention from the actual act of emulation. The distinction between the invention and the act of emulation is emphasized by the specification:

> The endian transformation method may be used as part of a complete emulation system. (page 8, lines 8-9)

Claims 1-4 recite "method[s] for emulating" that consist of a single step wherein bytes are "transformed". The recited method does not emulate a processor according to the meaning understood to those of skill in the art. The recited method transforms memory addresses, which may "*enable* software of one endian format to run on hardware of a different endian format," but does by itself not fulfill the definition of *emulate*. The Examiner addressed this in the previous Office Action by stating:

> The term "emulate" in claims 1-4 is used by the claim to mean "adapting program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words" [...] (previous Office Action, paragraph 17)

The Examiner maintains this analysis of the methods of claims 1-4. Additionally, Applicants' arguments in favor of the dictionary definition for *emulate* exemplify the indefiniteness of these claims. It is indefinite whether these are claims for methods of processor emulation, as argued by Applicants, or methods of transforming memory access addresses, at set forth by the claim limitations. It is indefinite whether the scope of these claims covers the act of *emulation* as known in the art or merely the act of transforming memory accesses.

Applicants' arguments have been fully considered but they have been found unpersuasive. The previous rejections of claims 1-4 under 35 U.S.C. § 112, second paragraph, are maintained.

Response to Rejections under 35 U.S.C. § 102

As an initial matter, Applicant is correct in presuming that claims 1-12 were rejected under 35 U.S.C. § 102. The Examiner apologizes for this inaccuracy and for any confusion it may have caused.

Regarding claims 1 and 9, Applicants argue primarily that:

> [...] Applicants' entire address space is a mirror image of the address space for the opposite endian format, whereas Loen only appears to create mirror images of bytes in specific words that are transformed. Accordingly, Applicant respectfully submits that Loen fails to teach or suggest transforming memory access addresses for an entire address space such that the aggregate pattern of bytes stored in memory is a mirror image of the distribution pattern of bytes which would have resulted if the memory was addressed by a processor of a type having a different convention for ordering the significance of bytes within words. Thus, each and every feature of amended claims 1 and 9 are not disclosed by Loen, and Applicant requests that the § 102 rejection of claims 1 and 9 be withdrawn.

The Examiner respectfully traverses this argument as follows.

Microsoft Computer Dictionary, Fifth Edition, provides the following definition:

> **address space** *n.* The total range of memory locations addressable by a computer.

The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, provides the following definition:

> **address space** **(1)** **(A)** The range of addresses that a computer program can access. *Note:* In some systems, this may be the set of physical storage locations that a program can access, disjoint from other programs, together with the set of virtual addresses referring to those storage locations which may be accessible by other programs. **(B)** The number of memory locations that a central processing unit can address. *See also:* virtual address space.
> **(2)** The memory locations that can be referenced by a process.
> **(3)** The memory locations that can be referenced by the threads of a process.

The disclosure fails to set forth which of these definitions is intended in the claims. The amendments to claims 1 and 9 therefore necessitate rejection under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicants' arguments related to the creation of a mirror image "of the address space" are therefore unpersuasive as it is unclear how to interpret this phrase.

Presuming definition **(1) (A)** from IEEE Standards Terms, Loen discloses creating a mirror image of the entire range of addresses that a computer program can access [Figs. 4A-4C; column 7, lines 47-61; implicitly disclosed is the functional code that operates on the *struct* shown].

Further, regarding Applicants' analysis of Loen, Applicants argue primarily that:

> [...] Loen only appears to create mirror images of bytes in specific words that are transformed.

The Examiner respectfully traverses this argument as follows.

Loen discloses that *"The reflection step may be performed in a variety of places, and is entirely mechanical and unrelated to the data element size being fetched"* (column 7, lines 12-14, emphasis added). The reflection step may therefore be performed on an entire address space.

8. Regarding claims 2 and 10, Applicants argue primarily that:

> [S]ince Loen appears to convert between endian formats on a word-by-word basis, bytes "p" and "q" would respectively be located in locations 8 and 23 in the little endian address space according to Loen's conversion process. From this, it can be seen that Loen fails to maintain [that 1) the offset between addresses of any two bytes stored in memory is unaltered and 2) the relative order of the addresses of any two bytes stored in the memory is reversed].

The Examiner respectively traverses this argument as follows.

Loen discloses that *"The reflection step may be performed in a variety of places, and is entirely mechanical and unrelated to the data element size being fetched."* (column 7, lines 12-14, emphasis added). Loen discloses that *"The first step is a reflection which must be performed on the bytes comprising the data double word or fragment thereof (see FIG. 3a)."* (column 7, lines 5-7). FIG. 3a discloses the properties indicated in Applicants' arguments and recited in the claims.

9. Regarding claims 3 and 11, Applicants argue primarily that:

> The Examiner asserts that [memory access addresses are transformed such that strings of bytes in the first endian format are stored [...] in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed] is inherent in Loen because "if the invention of Loen et al. did not aggregate byte strings as though they had not been translated, then could would need to be specially designed for the processor to accommodate this peculiarity of the endian transformation." Applicant traverses this reasoning and submits that Loen teaches quite the opposite effect. The disclosed system in Loen is designed to precisely handle such a peculiarity, because it converts between endian formats on a task-by-task basis so that it can dynamically change its endian mode (*see Abstract*). This dynamic conversion dictates that Loen performs conversion on a word-by-word basis, and, as shown above in the preceding examples, it is not inherent in Loen that it will store strings of bytes successively to aggregate in the same manner as the bytes would aggregate if the processor was of a different endian format and the memory access addresses were not transformed.

The Examiner respectfully traverses this argument as follows.

As an initial matter, the Examiner thanks Applicants for confirming that Loen converts between endian formats on a task-by-task basis, thus converting the entire address space (IEEE Standards, def. (2) or (3)).

The Examiner further thanks Applicants for bringing to attention that Loen does disclose the limitation as recited by the claims. Loen discloses that *"The first step is a reflection which must be performed on the bytes comprising the data double word or fragment thereof (see FIG. 3a)."* (column 7, lines 5-7). FIG. 3a discloses that a "string of bytes" are store successively in accordance with the second endian format. The Examiner apologizes for this oversight in the previous Office Action, however maintains that Loen anticipates the claimed invention.

10. Regarding claims 4 and 12, Applicants argue primarily that:

> Applicant develops this address transformation [A-B-L+S] to allow one extra arithmetic operation to be used with every load/store instruction to simply transform memory access address between two different endian formats.
> Contrarily, Loen discloses a two step process to convert between two different endian formats. [...] Loen fails to teach or suggest transforming a memory access address B of string length L to a new address using

> the simple arithmetic operation A-B-L+S, where A is the total number of bytes allocated to a program and S is the start address of the program.

The Examiner respectfully traverses this argument as follows.

As an initial matter, the expression A-B-L+S is not "one extra arithmetic operation" as would be understood by a person of ordinary skill in the art. The expression contains three arithmetic operators and produces a parse tree containing at least three intermediary results. A parse tree is a basic concept in the computer compiler arts that extracts the meaning from human-readable computer code and generates machine instructions. The application does not disclose a novel computer architecture wherein the expression A-B-L+S can be evaluated as "one arithmetic operation". Therefore, the meaning of this phrase is interpreted in light of the specification, wherein "one arithmetic operation" is generally interpreted according to "a single address transformation". Therefore, the observation that Loen discloses a "two step process" is irrelevant because, as shown by Applicants, the second step is "a single address transformation" (the XOR operation) and directly correlates to and anticipates that aspect of Applicants' invention.

Regarding the functional equivalence of Loen's address, the Examiner directs Applicants' attention to FIG. 4A and corresponding explanation (column 7, lines 48-65). The total memory allocated in FIG. 4A is A=8 bytes. The address of the byte being accessed is B=6. The start address of the program is S=0 bytes. The length of the data being accessed [defined as BYTE byte, (column 7, lines 50-55)] is L=1. As can be seen, the new offset after reflecting the memory is A-B-L+S = 8-6-1+0 = 1.

As another example, FIG. 4B shows the recalculated offset to access a halfword (2 bytes) after reflecting the memory. In this case, the address being accessed is B=4. The length of the data being accessed is L=1. The new offset as shown in FIG. 4B is A-B-L+S = 8-4-2+0 = 2.

Therefore, it has been thoroughly demonstrated that Loen does disclose and therefore anticipate the limitations of claims 4 and 12 which require that "the memory access address B of string length L is transformed to the address A-B-L+S." The Examiner notes that the claim does not recite that the method is limited to actually performing the arithmetic operations of A-B-L+S, but rather that the address is transformed to the address A-B-L+S, which Loen has been shown to anticipate.

11. Regarding claims 5-8, Applicants arguments are directed to independent claim 5 and refer back to the arguments for claims 4 and 11. Applicants' argue primarily that:

> This feature [modifying a referenced memory address using the operation A-B-L+S] is not taught or suggested by Loen, since Loen discloses a two-step conversion process including a reflection step and a separate XOR step.

The merits of this argument have been addressed *supra*.

12. The Examiner respectfully suggests that Applicants' review "The 8086 Microprocessor" by Kenneth J. Ayala regarding the definition of *word*. For the purposes of prosecution, the Examiner has interpreted phrases in the claims such as "[the] pattern of the bytes which would result if the memory was addressed by a processor [of the opposite endian format]" imply that certain trivial and obvious modifications are performed to allow the invention to operate properly regardless of the byte size of a word. Applicants' arguments bring that interpretation into

question where they attempt to distinguish between Loen's disclosure of words as opposed to Applicants' disclosure of bytes.

As would be understood by a person of ordinary skill in the art, Applicants' disclosure is directed to single byte words. To illustrate the trivial and obvious modifications to implement the method on multiple byte word architectures, Applicants' invention transforms the "memory access addresses", which addresses correspond to bytes on a single byte word architecture, but does not tamper with the contents of the words (bytes). In Loen, the invention transforms the "memory access addresses", which in certain embodiments, the addresses correspond to two bytes on two byte word architectures, but does not tamper with the contents of the words (two bytes). Both Applicants' invention and Loen transforms the memory access addresses but leaves the contents of those addresses unchanged regardless memory size represented by those addresses.

In the event that Applicants' arguments should be interpreted as meaning that the claimed invention rearranges the bytes within a word on a multiple byte word architecture, numerous rejections under 35 U.S.C. § 112, first paragraph, would be necessitated. For example, reading a memory access address for a multiple byte word wherein the bytes within the word have been rearranged would produce incorrect and inoperable data unless, by some undisclosed method, the bytes were returned to their original arrangement. Applicants' invention transforms memory access addresses, exemplified in the disclosure as byte addresses, but there is no teaching that the invention rearranges the contents of the memory access addresses, i.e. the contents of the words in memory. Loen therefore anticipates Applicants' invention on both single and multiple byte word architectures, as has been shown.

Applicants' arguments regarding the rejections under 35 U.S.C. § 102 of the previous Office Action have been fully considered, however they have been found unpersuasive.

Outstanding Rejections

Claim Rejections - 35 USC § 101

35 U.S.C. § 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

MPEP 2106 (IV)(B)(1) states:

> If the "acts" of a claimed process manipulate only numbers, abstract concepts or ideas, or signals representing any of the foregoing, the acts are not being applied to appropriate subject matter. Schrader, 22 F.3d at 294-95, 30 USPQ2d at 1458-59. Thus, a process consisting solely of mathematical operations, i.e., converting one set of numbers into another set of numbers, does not manipulate appropriate subject matter and thus cannot constitute a statutory process.

Claims 5-12 are methods that convert one set of numbers (memory access addresses in a first endian format) into another set of numbers (memory access addresses in a second endian format). As recited in claim 12, this is a method of simple arithmetic *"A-B-L+S"*, and anticipated by steps of human cognition given the broadest reasonable interpretation.

Compare to claim 13 for a "computerized method", which is a statutory process claim.

Claim Rejections – 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. § 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

13. Claims 1-4 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Where Applicant acts as his or her own lexicographer to specifically define a term of a claim contrary to its ordinary meaning, the written description must clearly redefine the claim term and set forth the uncommon definition so as to put one reasonably skilled in the art on notice that the applicant intended to so redefine that claim term. *Process Control Corp. v. HydReclaim Corp.*, 190 F.3d 1350, 1357, 52 USPQ2d 1029, 1033 (Fed. Cir. 1999). The term "emulate" in claims 1-4 is used by the claim to mean "adapting program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words", while the accepted meaning is "directly executing program code intended for a processor of a different design on the current processor." (See Microsoft Computer Dictionary) The term is indefinite because the specification does not clearly redefine the term.

Claims 1 and 9 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "address space" has several distinct definitions known in the art and it is unclear from the specification which definition is applicable. The metes and bounds of these claims are therefore indefinite. Some definitions for the term are provided below.

Microsoft Computer Dictionary, Fifth Edition, provides the following definition:

> **address space** *n.* The total range of memory locations addressable by a computer.

The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, provides the following definition:

> **address space (1) (A)** The range of addresses that a computer program can access. *Note:* In some systems, this may be the set of physical storage locations that a program can access, disjoint from other programs, together with the set of virtual addresses referring to those storage locations which may be accessible by other programs. **(B)** The number of memory locations that a central processing unit can address. *See also:* virtual address space.
> **(2)** The memory locations that can be referenced by a process.
> **(3)** The memory locations that can be referenced by the threads of a process.

Claim 13 is rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The preamble of the claim defines "a computerized method for enabling a computer program" and recites additional details regarding the program, but does not recite how or what the computer program is enabled to do. Presumably a word, critical to the meaning of the claim, has been inadvertently admitted. The Examiner presumes the method is for "enabling execution of a computer program".

Claim Interpretation

14. Regarding claims 1-4, the term "emulating a processor" has been interpreted as "adapting program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words."

15. Regarding claims 1 and 9, the term "for an entire address space" is interpreted according to IEEE Standards Definitions, **address space (2)**:

address space (2) The memory locations that can be referenced by a process.

Claim Rejections - 35 USC § 102

16. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

17. Claims 1-13 are rejected under 35 U.S.C. § 102(e) as being anticipated by US Patent No. 5,968,164 to Loen et al.

Regarding claim 1, Loen et al. discloses a method for adapting program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words wherein memory addresses for an entire address space are transformed to reflect the difference between big endian and little endian addressing conventions which result in a mirror image of the original bytes (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Regarding claim 2, Loen et al. also discloses that one addressing convention is the reverse of the other and the transformation preserves the offset between addresses and the relative order of addresses of bytes stored in memory is reversed (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Regarding claim 3, Loen et al. also discloses that byte strings stored successively by the processor aggregate in the same manner as if the code had been executed on a processor that is naturally biased toward the same endian convention as the code (column 6, line 20 – column 7, line 29; Figures 3A-3B). Loen discloses that *"The first step is a reflection which must be performed on the bytes comprising the data double word or fragment thereof (see FIG. 3a)."* (column 7, lines 5-7). FIG. 3a discloses that a "string of bytes" are store successively in accordance with the second endian format.

Regarding claim 4, Loen et al. also discloses that the memory address B of word length L is transformed to the address A-B-L+S where A is the total number of bytes allocated to the program and S is the start address of the program (Figures 4A-4D). Figures 4A-4D of Loen et al. discloses an example where A=8, S=0, and L is 8, 16, 32, or 64 bit, however also teaches other values for L (column 7, lines 62 – column 8, line 46).

For example, FIG. 4B shows the recalculated offset to access a halfword (2 bytes) after reflecting the memory. In this case, the address being accessed is B=4. The length of the data being accessed is L=1. The new offset as shown in FIG. 4B is A-B-L+S = 8-4-2+0 = 2.

Regarding claim 5, Loen et al. discloses a process for translating a program code instruction for execution on a programmable machine using a corresponding predetermined convention of ordering the significance of bytes within words of the address space (column 6, line 20 – column 7, line 29; Figures 3A-3B) comprising:

transforming the referenced memory address with respect to a fixed block size of memory in the programmable machine so as to change the referenced address value by an amount that is fixed for a given number of bytes being accessed in each word (Figures 4A-4D; column 7, lines 62 – column 8, line 46);

including the changed address reference in the output instruction so that there is no extra operation required during execution of the instruction to accommodate the convention for ordering bytes within words used by said predetermined programmable machine (column 7, lines 62 – column 8, line 46).

Regarding claim 6, Loen et al. discloses that the system is applied to program source code (column 7, lines 62 – column 8, line 46).

Regarding claim 7, Loen et al. discloses that a fixed block of memory is either big endian or little endian and therefore addressed from a predetermined one of its two ends depending upon the convention utilized for ordering the significance of bytes within the words (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Regarding claim 8, Loen et al. discloses that the translation causes a fixed block of memory contents for a big-endian machine to be inverted to the mirror image of that for a little-endian machine (column 6, line 20 – column 7, line 29; Figures 3A-3B).

Claims 9-12 all recite "an endian transformation system" which performs the methods of claims 1-4, respectively. As the invention disclosed by Loen et al. is indeed a system that performs a method used to reject claims 1-4 (column 4, lines 49-55), claims 9-12 are rejected for the same reasons used to reject claims 1-4 above.

Regarding claim 13, Loen et al. discloses a method for enabling execution of program code intended for a first processor to execute on a second processor which observes a different convention for ordering the significance of bytes within words (column 6, line 20 – column 7, line 29; Figures 3A-3B) comprising:

transforming the each memory address using a single transformation operation so as to change each memory access address to a value which would result if the memory were addressed by a processor of the second type (Figures 4A-4D; column 7, lines 62 – column 8, line 46).

Conclusion

Art considered pertinent by the examiner but not applied has been cited on form PTO-892.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp

Paul L. Rodriguez 7/27/05
Primary Examiner
Art Unit 2125